ASR Decoding in a Computational Model of Human Word Recognition

Louis ten Bosch, Odette Scharenborg

CLST, Radboud University Nijmegen, The Netherlands
{L.tenBosch,O.Scharenborg}@let.ru.nl

Abstract

Recently, a computational model of human word recognition, called SpeM, has been developed. In contrast to most current models of human word recognition, SpeM is able to process actual acoustic speech input, and decodes the incoming speech stream into lexical and non-lexical items. This model makes the links between HSR and ASR as explicit as possible. In this paper, we focus on unravelling the structure of the complex search space that is used in SpeM and similar decoding strategies. To that end, it discusses a number of properties of phone lattices in relation to canonical phone representations. Furthermore, we elaborate on the close relation between distances in this search space, and distance measures in search spaces that are based on a combination of acoustic and phonetic features.

1. Introduction

Both the research areas of automatic speech recognition (ASR) and human speech recognition (HSR) investigate the recognition process from the acoustic signal to a sequence of recognised units. For ASR, the target is to automatically transcribe the speech signal in terms of a sequence of items as close as possible to a reference transcription (e.g., [1], [2]). In HSR, the focus is on understanding how human listeners recognise spoken utterances. To investigate the mechanisms underlying the human speech recognition process, HSR experiments with human subjects are usually carried out in a laboratory environment. Based on the outcomes of these experiments, theories about specific parts of the HSR system are developed or refined. To put the theories to further test, they are implemented in the form of computational models for the simulation and explanation of HSR (e.g., Shortlist, [3], Trace, [4]).

One difference between ASR systems and most computational models of HSR is the representation of the speech signal at the input side. In most ASR systems, the input is (necessarily) the acoustic input itself, or a representation that can be derived from that acoustic input by an algorithm (e.g. a feature representation). Most HSR models, however, assume the presence of a handcrafted segmental symbolic representation of the speech in terms of prelexical units (see e.g., [3]). Recently, a computational model of human word recognition has been developed that circumvents the necessity of such a handcrafted representation. This model, named SpeM, makes use of techniques developed in the area of ASR [5]. It provides a successful and concrete demonstration of the computational parallels between HSR and ASR, by making the links between HSR and ASR as explicit as possible. SpeM decodes speech based on the theory underlying Shortlist; its implementation, however, is entirely different (see section 2).

The aim of this paper is to discuss aspects of the SpeM decoding in more detail. Since SpeM's ability to simulate data from human word recognition experiments is ultimately based on the structure of its search space, we will go into more detail concerning the complexity of this space, by elaborating on a number of properties of phone lattices in relation to canonical phone sequences. In SpeM and similar decoding approaches, the search space is determined by the interaction between acoustic scores of segments on the one hand, and penalties for phone insertions, deletions and substitutions on the other. Finally, we will show that the decoding can be linked to approaches in ASR that use phonetic features in combination with acoustic features. In order to put these issues in a proper perspective, we first give a brief overview of the SpeM model.

2. SpeM

The SpeM model is implemented as a multi-pass decoder (see Figure 1).

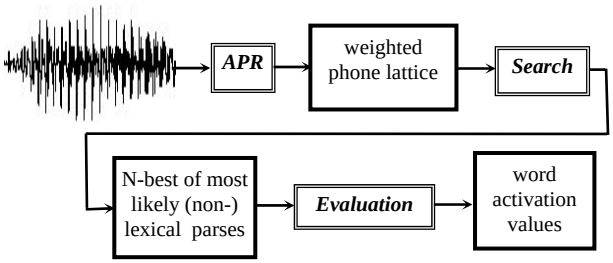

Figure 1. An overview of the implementation of the SpeM model (figure adapted from [5]).

In the first pass, an automatic phone recogniser (APR) processes the input speech signal and generates a (weighted) phone lattice. This lattice provides a probabilistic phone representation of the speech signal, and is input for the second pass which deals with the lexical search. Because the phone lattice is ultimately interpreted via the search algorithm, we will first pay attention to the search algorithm itself, before we discuss the search space (which is spanned by the phone lattice and the lexicon) in more detail in the next section.

The SpeM search module performs a search for sequences of lexical items such that the phonemic representation of these sequences (as determined by a vocabulary) is optimally matching the phone sequences in the lattice. The search is a node-synchronous Viterbi-like forward pass through a graph that is the product of the phone graph and the lexical graph (tree). This product graph is the actual search space. Each arc π in the product graph corresponds to an arc $\alpha(\pi)$ in the phone graph and an arc $\beta(\pi)$ in the lexical graph, and has a weight equal to the sum of the weights of $\alpha(\pi)$ and $\beta(\pi)$. The weight of $\alpha(\pi)$ is the acoustic score calculated by the APR; the weight of $\beta(\pi)$ consists of the unigram and bigram language model (LM) scores. The resulting hypotheses, i.e., paths

through the product graph, are considered in parallel; unlikely hypotheses are pruned away by a pruning mechanism.

A 'garbage' phone model is included in the lexicon, which can be mapped onto phones that do not belong to a lexical item. A (sequence of) garbage phone(s) is referred to as a non-lexical item. Furthermore, the search is able to deal with symbolic mismatches between phone sequences in the product graph and phone representations in the lexicon, by allowing (symbolic) insertions, deletions, and substitutions. Each type of mismatch has its own penalty, which can be tuned independently. Due to this flexibility, each parse may therefore consist of lexical items, word-initial cohorts (words sharing phone prefixes), non-lexical items, silence, and any combination of these (except that a word-initial cohort can only occur as the last element in the parse). The output of the search consists of an *N*-best list of hypothesised parses, each with its specific (acoustic and LM) cost.

An example of a search output (using orthographic representations for the sake of clarity) is provided below. The input is 'butter n brea', a reduced and truncated form of 'butter and bread'.

```
butter (150) *n* (200) brea* (300) (650)
but  (100) term (250) brea* (300)  (650)
but  (100) term (250) breath (350) (700)
```

Here *n* denotes a non-lexical constituent in the parse, required to make the first parse complete, and brea* denotes the word-initial cohort of phones shared by, e.g., 'bread' and 'breath'. The search provides scores of individual parse constituents as well as the accumulated score (between the brackets at the end of each line). The list of parses is updated and available after each node in the input phone lattice has been processed.

To complete this overview of SpeM, we finish with the evaluation module. In this module, the *N*-best list of parses is processed to generate, for each hypothesised word, a 'word activation' measure that varies over time. The concept of 'word activation' is used in HSR to indicate how easy a word will be for listeners to identify. Subjects hypothesise words based on the acoustic bottom-up match between speech input and internally stored representations of words (see [6]). Since the word activation measure is described elsewhere ([5], [7]), it will not be further discussed here.

3. The search space

As indicated above, the APR creates a weighted phone graph as a phonetic probabilistic representation of the acoustic signal. From the perspective of the search following the APR, an important issue is to what extent the phone graph must capture the phonetic detail in the signal in order to maximise the likelihood of containing phone sequences that correspond to lexical solutions. The basic assumption is that the APR is able to produce a phone lattice that encompasses a phonetic representation of the speech signal, including locally phonetically plausible variations, without being guided or constrained by lexical information.

What makes a phone graph a good phone graph in this context? Apart from evident factors such as the quality of the acoustic models and (implementation) details concerning splitting and recombination of arcs during the phone search, three factors have a decisive impact on the structure and contents of the resulting phone lattice: a) the phone insertion probability; b) the beam during the phone search by the APR; c) the use (and weighting) of a phone *N*-gram during the phone search. In combination with the symbolic mismatch penalties, these parameters fully determine the complexity of the search space.

For the search module to be able to find a lexical sequence with associated phone sequence P_c, there must be a phone sequence Q on a path through the phone lattice with the following property:

$$P_c = \text{argmin}_P\{\min_Q(score(Q) + d(P,Q) + LM(P))\} \qquad (1)$$

This expression is the mathematical formulation of the forward pass in the search. The term *score(Q)* is a shorthand for $-\log(P(X|Q))$, The signal X is given, P is hypothesised, and Q is a variable, running over the set of all paths available in the phone lattice. The term $score(Q)$ denotes the total path score of Q as defined by the phone lattice, while $d(P, Q)$ denotes the sum of all penalties for symbolic mismatches between the phone sequences P and Q. The final term $LM(P)$ denotes the language model score of the word sequence associated with P. Evidently, the minimising path Q depends on the hypothesised P.

Eq. 1 implies that the penalties for symbolic insertions, deletions, and substitutions are not free model parameters, but instead must be closely related to the distribution of path scores in the lattice. For example, let P denote a specific (arbitrary) hypothesis, and assume that for some path Q the term $d(P, Q)$ is made up by I insertions, D deletions, and S substitutions. In that case, the application of Eq. 1 has the very same effect as the evaluation of the score of the canonical path P in a *new* lattice *L'* that is obtained from the original lattice L by expanding *all* possible paths in L by allowing exactly I insertions, D deletions, and S substitutions with their appropriate costs. This new lattice *L'* (which does not physically exist, but is virtually constructed and explored during the search) depends on I, D, and S and, by construction, contains the sequence P. Repeating the same argument for any hypothesis P, this directly means that the *eventual* search space where *all* canonical sequences can be found is effectively the *union* of virtual lattices $L'(I, D, S)$ such that $I, D, S >= 0$. As a consequence, the entire distribution of the path scores in this union lattice is the union of the original score distribution H and shifted copies of this distribution: $\{H, H+cost(I), H+2*cost(I),\ldots, H+cost(I)+cost(D), \ldots, H+cost(D), \ldots, H+cost(S), \ldots\}$. And the only thing that really counts in the search is how 'far' in this union lattice any canonical phone sequences are alive. Since the structure of the union lattice is fully determined by L and by the symbolic mismatch costs, this means that the penalties for substitution, insertions, and deletions must be considered in relation to the structure of L, in particular to the distribution of paths in L that are canonical or almost canonical (i.e., with a small number of mismatches).

It is therefore of importance to know more about the score distribution of the phone lattice itself. To that end, we have examined a set of phone lattices from 669 utterances with read speech, selected from the Corpus Spoken Dutch (CGN, [8]). The phone lattices have been created using the HTK recogniser using acoustic monophone 3-state left-to-right HMMs with 8 gaussians/state, and using a phone zerogram.

The values for the phone insertion probability and the beam have been chosen such that the resulting phone lattices are phonetically plausible. First, the phone insertion probability was adjusted such that the resulting average number of phones in the best path was equal to the number of phones according to the canonical phone transcription defined by the reference transcription and the vocabulary (the resulting average number of phones per second is about 13). Second, the beam has been adjusted such that the time-averaged number of arcs with *different* phone labels is close to 3, i.e. a plausible number of realistic phonetic alternatives.

Given these choices, it appears that the number of *arcs* crossing a certain moment is on average 12 (minimum 2, maximum 48). The high number of local options implies that the number of paths through the lattice might be huge. Figure 1 confirms this. The figure shows the relation between the number N of paths in the lattice and the duration L of the utterance, approximately given by $^{10}log(N) = C * L$, with C equal to about 5.5. The constant C depends on the beam width and on the phone insertion probability: the larger the beam or the insertion probability, the larger C will be.

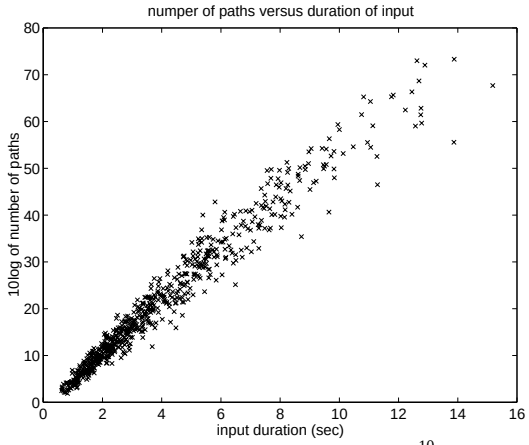

Figure 1. The figure shows the number (in ^{10}log, so e.g. 5 corresponds to 100,000) of paths in a phone lattice versus the duration (in sec) of the utterance.

From Figure 1 it is clear that for a phone graph corresponding to a stretch of speech of a few seconds, even reasonably long *N*-best lists of phone sequences (of, say, length 50,000) capture only a negligible fraction of the information in the graph. An *N*-best list is interesting because it captures local information about probabilistic segmentation (e.g., [9]), but it has hardly any relevance for capturing the canonical sequence (actually, the probability that the *complete* graph contains the canonical correct phone sequence decreases rapidly with the length of the utterance, and is for our data set smaller than 1 percent for utterances longer then 1.5 sec). Much more relevant for the structure of the search space in SpeM-like decoding is the minimum number of substitutions, insertions, and deletions required to construct the canonical sequence from a path through the phone graph. Table I shows this number (the minimum Levenshtein distance) as a function of the utterance duration for the 669 utterances. The first column refers to the duration category, the second column presents the total number of utterances per category, while the third column contains the minimal Levenshtein distance, averaged over all utterances in the category. A comparison between this number and the duration shows that the canonical path is about two repairs per second away from the best matching solution in the graph. Given that the canonical path contains 13 phones/sec, on average 1 out of 6 phones must be 'repaired'. The fourth column presents the average location of the best matching path in the phone graph, expressed in percentiles of the entire score distribution of paths in the phone graph. '0' means the cheapest path, '10' means at the 10th percentile, etc. The resulting best matching path has always been found in the top 7 percent of the paths, arranged by their score.

Table I. The minimum Levenshtein distance as a function of the utterance duration.

Duration cat. (sec)	*#utt*	*Average Levenshtein distance*	*Location of found path (percentile)*
0.50-0.75	6	1.2-1.4	<5
0.75-1.0	19	2.2	<7
1.0-1.5	54	2.5	<6
1.5-2.0	73	3.3	<4
2.0-3.0	150	3.6-4.2	<6
> 3.0	367	> 4.1	-

Figure 2 shows in another way how the information in *N*-best lists is only of marginal value for finding complete sequences. It shows the number of different phone sequences among the 5,000-best as a function of the duration of the 669 utterances. Silence arcs have been discarded. As expected, for longer utterances, all phone hypotheses in the 5,000-best list tend to be unique. The 'hockey stick effect' for low durations is due to the fact that short utterances relatively contain more silence than longer utterances which evidently reduces the number of different phone paths.

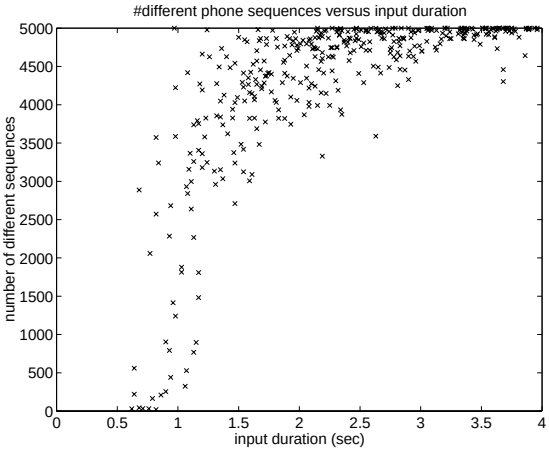

Figure 2. This figure shows, for 669 utterances, the number of *different* phone paths in the 5,000-best list as a function of the duration of the utterance.

The final observation that we want to make is about an interpretation of Eq. 1 that enables us to make a close link with phone decoding strategies that are based on signal feature representations augmented with phonetic features. Minimising the right-hand side of Eq. 1 can be thought of as looking for a path $Q = \{q1, q2\ldots\}$ in such a way that it optimally matches X (by minimising $-log(P(X|q))$) and at the

same time minimises $d(P,Q)$. The resulting alignments between the speech frames {*x1, x2, …*}, the phones in *Q* {*q1, q2, …*}, and in the canonical phone sequence *P* {*p1, p2, …*} are schematically shown in Figure 3 (top). However, $d(P, Q)$ is a sum of local symbolic distances between {*p*} and {*q*}, a sum which can be represented by the sum of distances between symbolic phonetic feature vectors. Furthermore, the alignment between *X* and *Q* implicitly assigns to each frame in *X* a phonetic representation inherited from the phones {*q*}. So we can rewrite the sum of *score*(*Q*) and $d(P, Q)$ in Eq. 1 as one *single* distance between two *augmented* sequences: one sequence *augX* of feature vectors {*x*} augmented (via the alignment) with phonetic features from {*q*}, and a sequence *augP* of {*p*} augmented with their own phonetic features (Fig. 3, bottom displays the new situation).

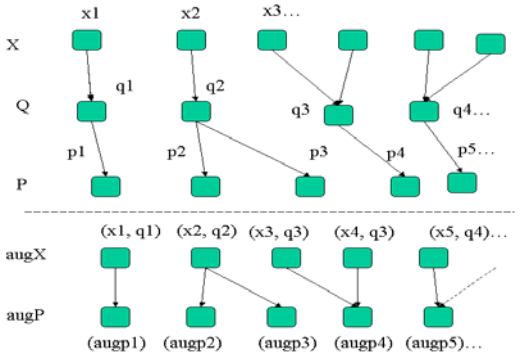

Figure 3. Top: Association between speech frames {*x*}, phone sequence {*q*} and {*p*} by alignment via Eq.1. *Bottom*: The same association, with one single distance between augmented representations.

This implies that the search for lexical parses in the phone lattice can be interpreted as a search for a match between an augmented representation of the frames in *X* and an augmented representation of the segments in *P*. The correspondence is not always exact, since in Eq. 1, the minimising *Q* is dependent on *P*, while here it is assumed that each frame in the speech signal can be assigned a static phonetic feature representation. But we know from other research (e.g., [10], [11]) that such a feature assignment can be done with reasonable plausibility. Furthermore, although the number of different paths in the phone lattice may be large, the *local* variations are mostly within one phonetic class. This means that speech recognition approaches based on combinations of acoustic and phonetic information in the search can be linked in a natural way with a SpeM-like speech decoding. It also shows how the *symbolic* penalties and *acoustic* scores can be brought into one framework.

4. Conclusions

The search space in SpeM and similar decoding techniques has been studied by considering a number of properties of the phone lattice. The search space can be regarded as the union of the original phone lattice and virtual lattices that are related to symbolic insertions, deletions, and substitutions. The penalties for symbolic mismatches are closely related to the distribution of (near-) canonical paths in the lattice. Phone lattices built with a phone loop with zerogram phone-LM and plausible values for beam and phone insertion penalty show that the probability of observing the correct phone sequences decreases rapidly with the length of the utterance. In order to be able to find the correct lexical solution, the flexibility to deal with the symbolic mismatches between the sequences from the lattice and the canonical phone sequences is absolutely essential. Given the canonical correct phone path, the best matching path through the lattice has always been found in the top 7 percent of all paths, and the required minimum number of repairs (substitutions or insertions or deletions) was found to be about 2 per second. This result is based on an analysis of 669 recordings of read speech.

Finally, we have indicated the close resemblance between the lexical search in SpeM on the one hand, and the approaches in ASR using phonetic features on the other. This relation opens possibilities to bring acoustic/phonetic approaches in ASR and the computational modelling of human speech recognition in a more unified paradigm.

5. Acknowledgements

The first author participated in the FP5 project COMIC (nr. IST-2001-32311). Annika Hämäläinen provided the CGN data and the HTK acoustic models.

6. References

[1] Rabiner, L., Juang, B.-H., “Fundamentals of speech processing”. New Jersey: Prentice Hall, 1993.

[2] Jelinek, F., “Statistical methods for speech recognition”. Cambridge, MA: MIT Press, 1997.

[3] Norris, D., “Shortlist: A connectionist model of continuous speech recognition”, *Cognition*, *52*, 189-234, 1994.

[4] McClelland, J.L., Elman, J.L., “The TRACE model of speech perception”, *Cognitive Psychology*, *18*, 1-86, 1986.

[5] Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M., “How should a speech recognizer work?”, *Accepted for publication in Cognitive Science.*

[6] McQueen, J.M., Speech perception, In K. Lamberts, R. Goldstone (Eds.), *The handbook of cognition* (pp. 255-275). London: Sage Publications, 2004.

[7] Scharenborg, O., ten Bosch, L., Boves, L., “‘Early Recognition’ of Words in Continuous Speech”, *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, US Virgin Islands (CDROM), 2003.

[8] Oostdijk, N., “The design of the Spoken Dutch Corpus”. In Peters, P., Collins, P., Smith A. (Eds) *New Frontiers of Corpus Research* (pp. 105-112). Amsterdam: Rodopi, 2002.

[9] Lee, S., Glass, J. “Real-time probabilistic segmentation for segment-based speech recognition”, *Proc. ICSLP*, Sydney, Australia. pp. 1803-1806, 1998.

[10] King, S., and Taylor, P. “Detection of phonological features in continuous speech using neural networks”, *Computer Speech and Language*, *14*(*4*), pp. 333-353, 2000

[11] Livescu, K., Glass, J., “Feature-based pronunciation modeling with trainable asynchrony probabilities.” *Proc. ICSLP*, Jeju, South Korea, October 2004 (CDROM).